Day-7
07-04-2026

Style Binding [style.propertyname]

- We are binding red color to the style binding

Syntax:- [style.protertyname] = " 'value' "

app.component.ts

```
import {Component} from '@angular/core';
@Component({
 selector: 'app-root',
 templateUrl: './app.component.html',
 styleUrls : ['./app.component.css']
})
export class AppComponent{
  boolval = true
   title = 'style Attribute'
}
```

app.component.html

```
<h1 [style.color]="'red'">style Binding</h1>
<h2 [~~color~~ style.color]=" boolval ? 'blue' : 'red' ">
       style binding using boolean value </h2>
```

Output:-

Style Binding

Style Binding using boolean value

With the help of style binding we can apply one color at a time. If you want to apply multipli css property then we have to go for [ngStyle]

ngStyle:-

why ngStyle: In order to apply multiple css properties we have to use style directive

* Syntax : [ngStyle]
* [ngStyle] = to this we can bind any of the object()
  example: literal
* The [ngStyle] directive lets you to set a given DOM element style properties.

app.component.ts

```
import {Component} from '@angular/core';
@Component({
  selector: 'app-root'
  templateUrl: ---
  styleUrls: --
})
export class AppComponent{
applyStyle = {
color: "yellow",
backgroundColor: "red",
fontStyle: "italic"
}
title = "styleAttributes";
}
```

app.component.html

assigning variable

```
<h2 [ngStyle]="applyStyle">
   styling using ngstyle</h2>
```

Output:-

color: yellow

$\boxed{\text{styling using ngstyle}}$

italic

bgcolor: ~~yell~~ red

class binding:

* To the element we can apply class attribute (class="one")
  To the element we can apply class binding([class]="value")

app.component.ts

```
import {Component} from '@angular/core';
@Component({
 selector: 'app-root'
 templateUrl: './app.component.html',
 styleUrls: ['./app.component.css']
})
export class AppComponent{
 classvariable="two"
 title='style Attribute';
}
```

app.component.css

```
.one
{
 color:red;
}
.two
{
 color: green;
}
```

app.component.html

```
<h2 class="one">class binding </h2>
<h2 [class]="classvariable"> class
        binding </h2>
```

Output:-

Class binding → red color

class binding → Green Color

- If we use both class attribute and class binding, class binding will be given more importance. class attribute becomes dummy attribute.
- You can either use class attribute or class binding but not both.

app.component.ts

```
import{Component} from '@angular/core';
@Component ({
 selector: app-root;
 templateUrl: './app-component.html';
 styleUrls: ['./app.component.css']
})
export class AppComponent{
classvariable="two"
title="style Attributes';
}.
```

app.component.html

```
<h2 [class]="classvariable" class="one"> class binding </h2>
```

app.component.css

```
.one
{
  color: red
}
.two
{
  color: green
}
```

Output:

Class binding $\rightarrow$ color: green.

attribute binding [attr.attributename]

- Whenever we invoke attribute binding then we have to make use of "attr"

  Syntax: [attr.attributename] = variable name.

app.component.ts

```
import{Component}from '@angular/core';
 @Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css'],
 })
 export class AppComponent
 {
    colspanValue=2;
title='style Attribute';
 }
```

app.component.css

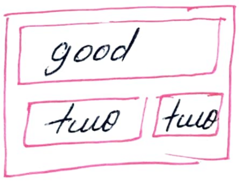
```
table, td, tr {
        border: solid;
    }

td{
   background-color: red
}
```

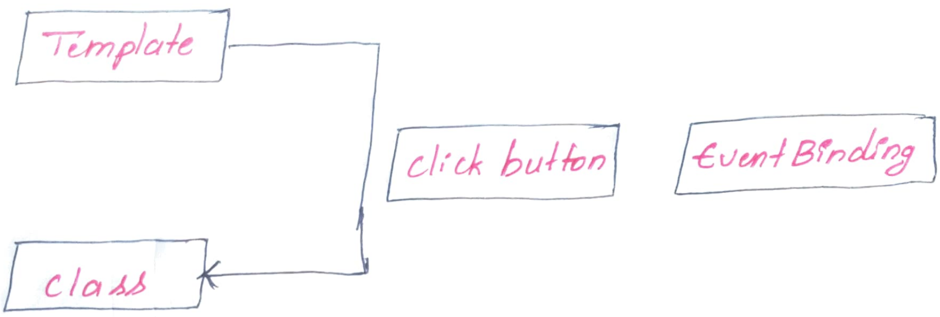
app.component.html

```
<table>
   <tr>
     <td [attr.colspan]="colSpanValue">good</td>
   </tr>
   <tr>
     <td> two</td>
     <td> two</td>
   </tr>
</table>
```

Output



Event Binding (Event)



* In data binding the flow was from component class to component template.
* Sometimes we have to respond to user event such as mouse clicks or keyboard clicks or keyboard events. In these cases the data will be flowing from template (html) to component class (ts file).
* Whenever click event has to be returend in the html page we make use of "( )"

Syntax :- (event)

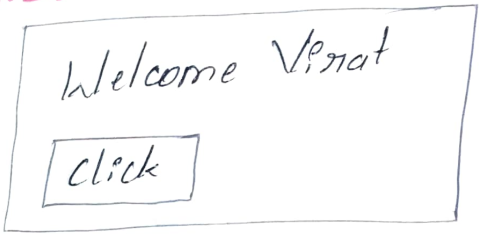
app.component.ts

```
import {Component} from '@angular/core';
@Component ({
 selector: 'app-root'
templateUrl:
 styleUrls:
})
export class AppComponent{
title = 'styleAttribute';
name = "virat"
clickfun(){
alert ('you have clicked button')
}
}
```

app.component.html

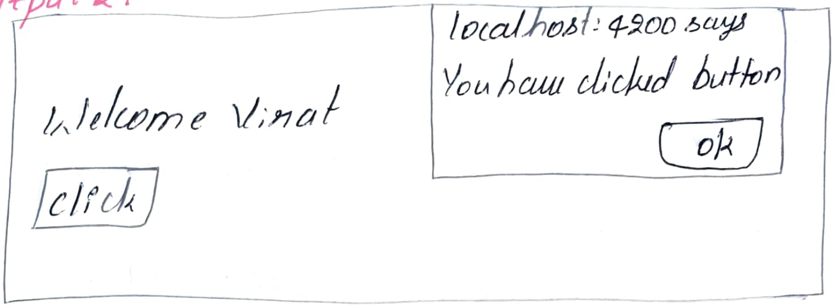
```
<h2>Welcome {{name}} </h
<button (click) = "clickfun()">
   click </button>
```

Output 1:-


Welcome Virat

click

Output 2:-


Welcome Virat

click

localhost:4200 says

You have clicked button

ok

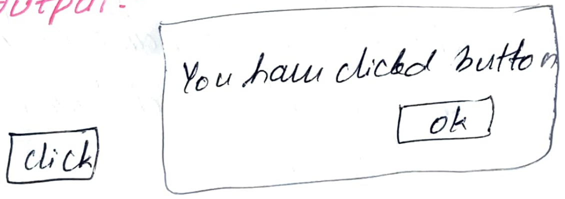
app.component.ts

```
export class AppComponent
{
    name = "virat",
    clickfun(event)
    {
        alert(" You have clicked a button")
        console.log(event)
    }
   title = "styleattribute'
}
```

app.component.html

```
<button (click)="clickfun(event)">click</button>
```

Output:-


click

You have clicked button

ok

Console screen

MouseEvent{isTrusted: True, screenX: 34, screenY: 229, clientX: 34, clientY: 158}